INF 558: Building Knowledge Graphs

Project Summary: Marvel and DC-Comics comic-books KG

Basel Shbita shbita@usc.edu
Ke Xu kxu582@usc.edu

November 29, 2018

1 Introduction

Comic Books are more than just superhero stories, they are a pop culture phenomenon, they give hope and encourage readers to think differently!
In this project, we planned, designed, implemented and deployed a knowledge graph which captures Marvel and DC-Comics brands comic-book characters, comic-book issues and movies.
Our main goal is to enable users to interact with the KG we constructed and explore the universe of the biggest comics-domain leaders. Additionally, we would like to use existing ontologies as much as possible to allow easier deployment of our KG and link it to external resources for easier user interaction.
The system allows any user to analyze data and become familiar with the domain, including top writers, inkers, character-creators, movies and origins (i.e. powers, species, etc..). Our UI allows any user to easily navigate the KG and obtain insights for any question in this domain. The system is a catalyst for anyone who wishes to enter the world of comics, as a consumer, provider or just out of interest!
To achieve our goal, we have defined an architecture which is depicted in Figure 1.

2 Challenges

Our first milestone was to crawl data using *Scrapy*. To allow a more focused process we generated all the available hub pages to easily reach all required authority pages. The biggest challenge in this step was to discard irrelevant pages, we faced more than 1M of pages which included URL-per-user-comment; we had to implement strict pattern matching and re-crawl the data more than three times to collect a satisfying set of pages.
For the task of data extraction we used *BeautifulSoup* for structured data and manually created some in-code-rules for unstructured data. For example, we have ran into some cases where date fields were not of a valid format. In such cases, we implemented a rule-based system to allow a proper extraction of data.
Our next challenge was to perform data cleaning, normalization and curation. In this stage we had to deal with inconsistent field formats: date fields used numerous different formats which we had to discover, classify and perform the proper transformation to match the format we want to use.
Another task we had to handle was the definition of a custom ontology that captures the concepts that will best describe each resource while utilizing existing ontologies as *schema.org, xsd, foaf and dbpedia (dbp, dbo, dbr)*. This process took a couple of iterations until we were satisfied with the result.
The generation of the KG triples was done using the *rdflib* python library and consisted of many iterations of data materialization which was used later for entity resolution and linking. To perform the ER, we used the *ISI-RLTK* tool and implemented several RLTK records, entity-matching functions and similarity measures. An example of such flow is presented in Figure 2. Results of the entity matching are shown in Table 1.
Our biggest innovation in the system was introducing the colors associated to each character which we extracted using a tool we implemented from scratch (using *Scipy* and *Numpy*). The flow included downloading images, analyzing pixels, clustering to known-colors by measuring Euclidean distance and generating proper color triples.
The system UI was a big challenge as well, we implemented a *Flask* application which constructs the proper SPARQL query to capture the users input via a web-form and generates a user-friendly response from the results using *Chart.js* and *Bootstrap* frameworks. In addition, we offer a dbpedia-like navigation, manual query construction and a dedicated search-within-results feature.
Finally, we had to validate in each step the correctness of our KG and to make sure that each problem we allegedly-fixed was properly handled. This was done via a set of SPARQL queries that we manually designed to profile the problematic issues we encountered. We also used online RDF validators and visualization tools for additional verification.
To properly evaluate our system and determine the success of our project, we have asked for feedback from colleagues and friends whom we asked to use our system. The users responded positively to each component in the system.

3 Conclusions

We have applied many different techniques which are fundamental and valuable in tackling problems one faces when building a knowledge graph that is based on real-world data. Each component in the system is vital and should be designed and planned carefully!
First, when it comes to crawling, one should consider the different techniques for focused crawling and accurately discard irrelevant pages, this could save precious time.
Second, when designing an ontology, one should accurately capture relationships between classes and properties. Working with existing ontologies can make it much easier.
Third, we believe that a good indicator for a well-constructed-KG is the complexity of composing a SPARQL query for any request. We have invested many efforts in the ontology definition and triples generation, which made the design part of the front-end-request translation to a back-end-query very straight-forward and access-efficient.
Moreover, one cannot solely rely on the system output to properly profile, recognize and handle problems in the system. The scale is too big and each iteration takes a lot of time to perform. Therefore, one should properly plan and implement ahead of time the different components which are relevant for the identification of errors and their curation. We have encountered this kind of problem during the triples generation stage where we faced not-so-common formats of common attributes like dates, places, physical-attributes (eyes, hair) and etc...
Additionally, when performing entity resolution, utilizing the *RLTK* blocking schemes is extremely useful and can save valuable computation time.
Last, we believe that a KG based data is more easy to navigate, explore and capture knowledge. Many relational databases do not give the option of such navigation. In the domain we chose, and to the best of our knowledge, there is no active system which gives a user an easy way to navigate between the different personas behind the comic-book industry. Our knowledge graph makes it easy for a user to explore top comics-book inkers, writers and producers with any user-defined constrains.
The link to our video tutorial is https://youtu.be/MYFi45EeZV0

A Appendices

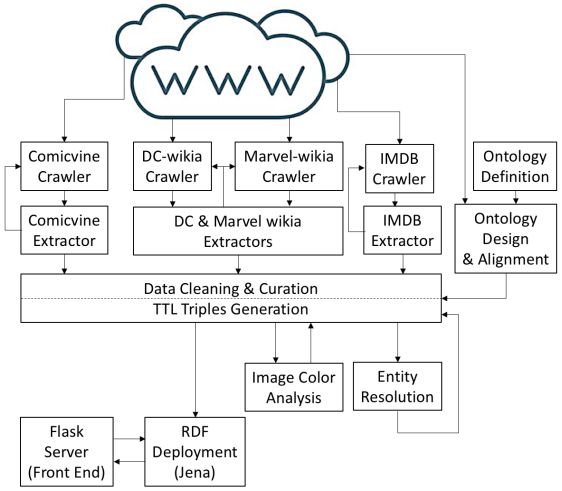

Figure 1: The Project Architecture

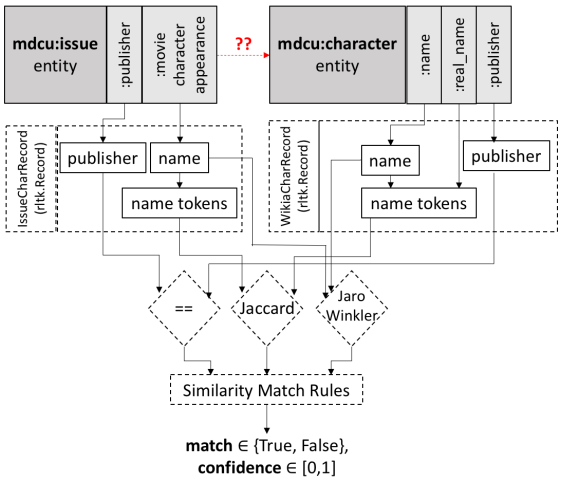

Figure 2: One Example of an Entity Linking flow using RLTK

Table 1: Entity Linking Statistics

Type	C1	C2	Match
Movie->Char	763	32165	206
Issue->Char	10796	32165	2909
Issue->Location	1247	3919	652
Issue->Team	1235	7401	562